

Prove di ruota MQC in js8call

Dopo gli [esperimenti di Mercoledì](/stream/2022/05/29/js8call/) con js8call, la voglia di provare qualcosa di nuovo era ancora forte. Il giorno dopo, Giovedì 26, è il turno di IU2OMT Paolo e IW1DUS Mario.

Callsign	TX	RX	Nome	Time off
IU2OMT	-08	+13	Paolo	2042
IW1DUS	-15	-02	Mario	2048

Segnali come sempre interessantissimi per i nostri 5W. Facciamo quattro chiacchiere poi propongo di testare una funzionalità di js8call chiamata [group callsign](https://docs.google.com/document/d/159S4wqMUVdMA7qBgaSWmU-iDI4C9wd4CuWnetN68O9U/edit#heading=h.p2czxqrknaod) [https://docs.google.com/document/d/159S4wqMUVdMA7qBgaSWmU-iDI4C9wd4CuWnetN68O9U/edit#heading=h.p2czxqrknaod]. Aggiungiamo @MQC alle group callsign del nostro applicativo e tentiamo un QSO di gruppo.

The screenshot displays the JS8Call v2.2.0 interface. At the top, the frequency is 7.078 MHz (1648 Hz offset) and the current call sign is IU5BON - JN530T. The interface shows a list of messages in a central pane, including a group call message from @MQC: "CONDIZIONI. PERDO PACCHETTI. ERA ANDATO FUORI DI 600 HZ". To the right, a list of callsigns is shown, with @MQC highlighted. The bottom part of the screen features a waterfall plot showing the frequency spectrum from 1000 to 3500 Hz. The status bar at the bottom indicates "Decoding ft991a JS8 Last Tx: @MQC DIT DIT".

La cosa funziona, ma comunque richiede un po' di pratica. Approfondendo il manuale sembra che esistano delle group callsign ufficiali che sono ottimizzate per la trasmissione

come @QRP , @SOTA , e @POTA .

La prossima volta sarebbe meglio usare @QRP al posto di @MQC per sfruttare le ottimizzazioni di js8call.

73 de IU5BON

2022-06-02

<https://l0g.in/5K5HPU>